

Diseñando interfaces humanas: Librería gráfica Microchip's y Magic Touch Capacitive Sensing



Agenda

- Sistema LCD básico
- Microchip Arquitectura de Librería Gráfica
- Fundamentos del touchscreen de 4 cables
- Filosofía del Mtouch
- Descripción de Herramientas
- Descripción de la interfaz MTouch
- Opciones de Touch Proporcional
- Opciones de presión apareada y sleep



Sistema LCD Gráfico



Graphical Display Market

- Sistemas POS y terminales de mano
- Equipamiento de testeo industrial y de mano
- Scanner de código de barras
- Aplicación en paneles frontales












Systema LCD Gráfico



Resolución y tamaño del Buffer de imagen

Monochrome

	X resolution	Y resolutio	Pixels	Bits	Bytes	Interface
1/16 VGA	160	120	19,200	19,200	2,400	4 Data
1/8 VGA	240	160	38,400	38,400	4,800	4 Data
QVGA	320	240	76,800	76,800	9,600	4 Data

Color STN

	X resolution	Y resolutio	Pixels	Colors	Bits	Bytes	Interface
1/16 VGA	160	120	19,200	8	57,600	7,200	8 Data
1/8 VGA	240	160	38,400	8	115,200	14,400	8 Data
QVGA	320	240	76,800	8	230,400	28,800	8 Data

Color TFT

	X resolution	Y resolutio	Pixels	Colors	Bits	Bytes	Interface
1/16 VGA	160	120	19,200	4,096	230,400	28,800	12 Data
1/8 VGA	240	160	38,400	4,096	460,800	57,600	12 Data
QVGA	320	240	76,800	4,096	921,600	115,200	12 Data
1/16 VGA	160	120	19,200	262,144	345,600	43,200	18 Data
1/8 VGA	240	160	38,400	262,144	691,200	86,400	18 Data
QVGA	320	240	76,800	262,144	1,382,400	172,800	18 Data



LCD Gráfico – Chip sobre el Vidrio (COG) Diagrama en Bloques



Interfaz Display RGB con Controlador Gráfico Diagrama en Bloques

PIC24F
PMP
Libraries:
Line, Circle, Rectangle, 3D Objects etc
CS
RD/ WR
EN
A0
DB0 – DB7
RST
Digital Control Logic
Image Buffer (RAM)
Graphic Accelerator (Optional)
Graphics Controller
Gate Driver
LCD Glass (Monochrome, STN, TFT)
LCD Module
Back light
RGB Interface



Requerimientos del sistema

- MCU PIC16- or 32-bit
- Interfaz PMP (Parallel Master Port)
- 2 Canales ADC y 2 Ports I/O Digitales
  - Requiere soporte para touchscreen 4 cables
- Librerias Gráficas
  - Microchip Graphics Library v1.3 o mayor
  - Ramtex – GUI Lib (www.ramtex.dk)
  - Segger – emWIN (www.segger.de)
- Display Gráfico QVGA or WQVG



Graphics PICtail™ Plus Demo Board



- **Trabajar con ambos PIMs 16- y 32-bit**
- **Seleccione el jumper:**
  - **Instalar el módulo LCD con controladorconstruido OR**
  - **conector estilo RGB y controlador SSD1906**



Arquitectura de la Librería gráfica de Microchip



Características de las librerías Gráficas de Microchip

- Soportan MCUs PIC de16- and 32-bit
- Librerías con licencia libre
  - Incluyen código fuente
  - soportan Touch screen
- Diseño Orientado a Objeto
- Entorno de desarrollo completamente equipado de bajo costo
  - Explorer 16 ($129.99 USD)
  - Graphics PICtail Plus Paca hija ($135.00 USD)



Descripción de la Librería

Application Layer

Mouse | Key Pad | Touch Screen

Graphics Objects Layer

Message Interface

Draw Function 1 (i.e. 3D Button)

Draw Function 2 (i.e. Progress Bar) …

Draw Function N

Graphics Primitive Functions
(Non-accelerated Line, Circle, Bar, OutText etc)

Graphics Primitives Layer

Display Device Driver (DDD)
(i.e. PutPixel, GetPixel)

Graphical Accelerator
(Optional)

Device Driver Layer

Graphics Display

Aplicación Específica

FREE

Módulos Genéricos

Dispositivos Específicos



Drivers de la Librería de Microchip

- **LCD Module Controller ICs:**
  - **Densitron: HIT1270**
  - **LG: LGDP4531**
  - **Renesas: R61505U**
  - **Orise Tech: SPFD5408A** *(new)*
  - **Samsung: S6D0129, S6D0139**
  - **Solomon Systech: SSD1339**
- **Controlador LCD externo:**
  - **Solomon Systech: SSD1905, SSD1906** *(new)*
- *Pronto existirán mas…*



Arquitectura de la Librería Gráfica Opción del Tiempo de Compilación

- Configuración con Bloqueos (DEFAULT)
  - Flujo lineal simple
  - Control directo cuando el Display es refrescado o rediseñado
- Configuracion sin bloqueos
  - Utiliza la representación basada en estado de objectos
  - Uso eficiente del tiempo del procesador.
- Opciones seleccionadas en: *GraphicsConfig.h*



Primitivas Funciones para representación

Funciones de control

- *InitGraph()* – Realiza la secuencia de reset apropiada de los dipositivos para inicializar los registros de control de los mismos.
- *ClearDevice()* – Borra el display por rellenando el display con el color de fondo corriente.
- *SetLineType()* – Setea el tipo de lineas (Solido, Punteado, Estrellado). Todas las líneas rendidas utilizarán este tipo.
- *SetColor()* – Setea el color del siguiente objeto primitivo para ser rendido.
- *SetFont()* – Setea la tabla de fonts para ser usada cuando declare el texto en el display.



Funciones Primitivas de la representación

- *Line(,,,,)* – **declara una linea de un tipo seteado por** *SetLineType()* **y seteo del color por** *SetColor()*
- *Bar(,,,,)* – **declara una barra con seteo del color por** *SetColor()*
- *Rectangle(,,,,)* – **declara un rectangulo usando el color corriente**
- *OutText(,,,)* – **declara un string de texto de un caracter a un timempo usando el font corriente de la tabla seteada.**
- *PutImage(,,,)* – **declara una imagen o bitmap**
- *Circle(,,,)* – **declara un circulo usando el color corriente seteado**
- *DrawPoly(,,,)* – **declara un polígono usando el color corriente seteado**
- *FillCircle(,,,)* – **declara un círculo relleno usando el color corriente seteado**



Capa de objetos graficos

Objetos Gráficos Avanzados (Aparatos)

- Boton
- Ventana
- Caja de Cheques
- Caja de grupo
- Boton de radio
- Texto estático
- Imagen
- Caja de edición
- Caja de Lista
- Pote deslizable
- Barra tipo scroll
- Barra progresiva
- Medidor
- Dial
- y mucho mas….



Capa de objetos Gráficos

Barra Scroll

Medidor

Caja de chequeo

Rounded Button

Boton circular

Imagen



Objetos Gráficos Avanzados

- Control del objeto alcanzado vía APIs
- Comportamiento de un objeto determinado y estado actual de las acciones del usuario
  - Update o no?
  - Mensaje o no?
- Listado de linked usado para manejar Objetos
  - Permite Listas Multiples
  - La lista activa recibe los objetos creados
- El esquema del estilo define la acción de la representación
  - Tablas de Fonts y colores
  - Esquemas por Default o a medida

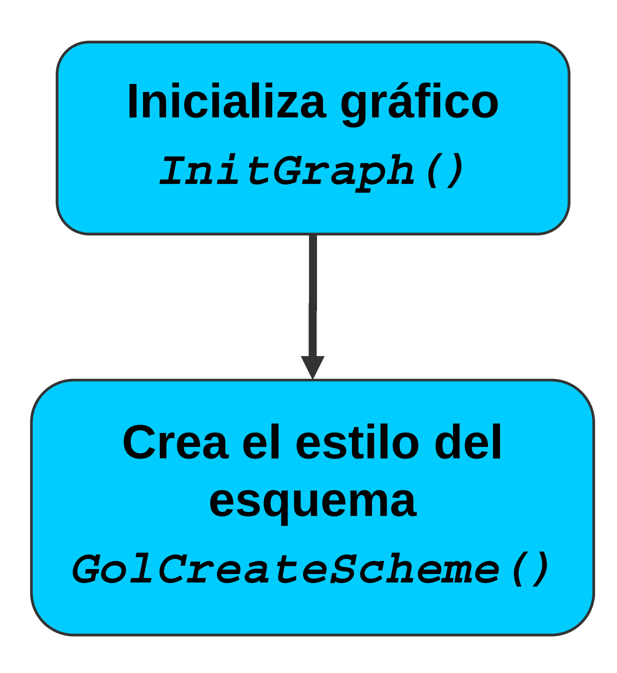


Objetos Gáficos Avanzados

- *objCreate()* –
  - Crea objetos con parametros para setear dimensiones de estos, localización e estado inicial
  - Suma el nuevo objeto a un listado de linkeado Global
- *objDraw()* –
  - Presenta el objeto sobre el display
  - Llama a refrescar el objeto siempre que el estado del objeto cambie
- Manejo de API's –
  - Maneja el estado y apariencia del objeto
  - Habilita o desabiñita un objeto
  - Cuando desabilita, todos los mensajes para el objetos son ignorados.
- Suporte para la destrucción de un objeto



Uso de la Librería Gráfica

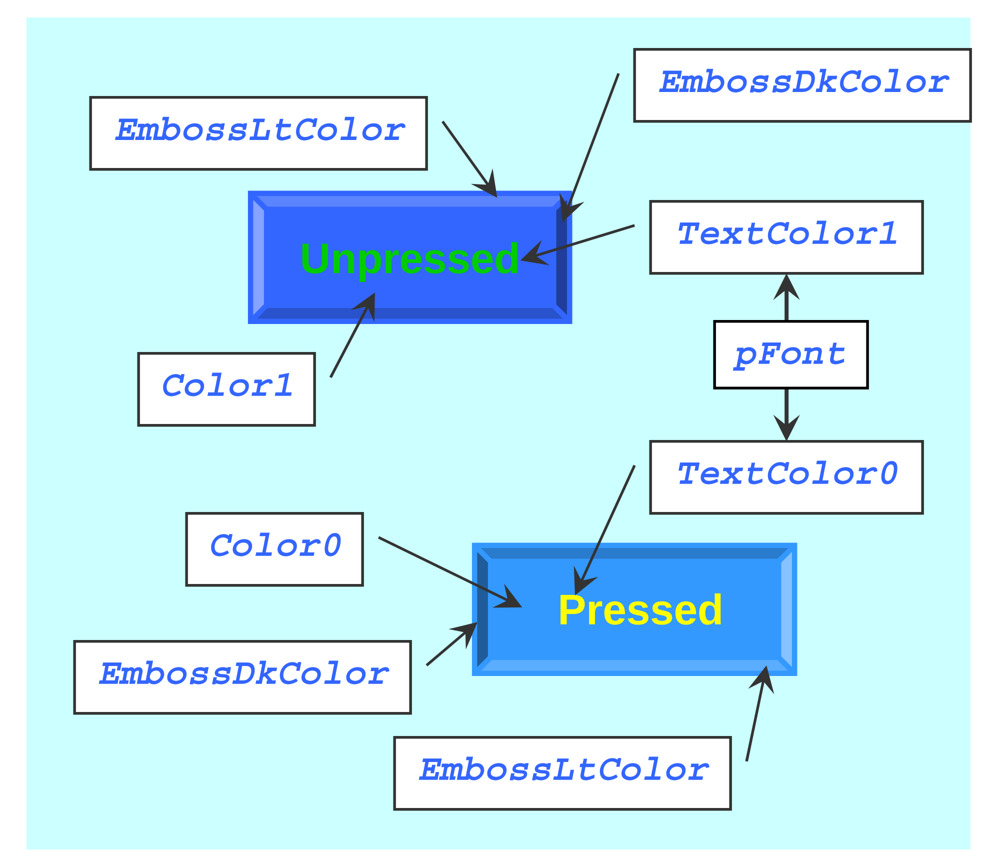




Estilo del Esquema

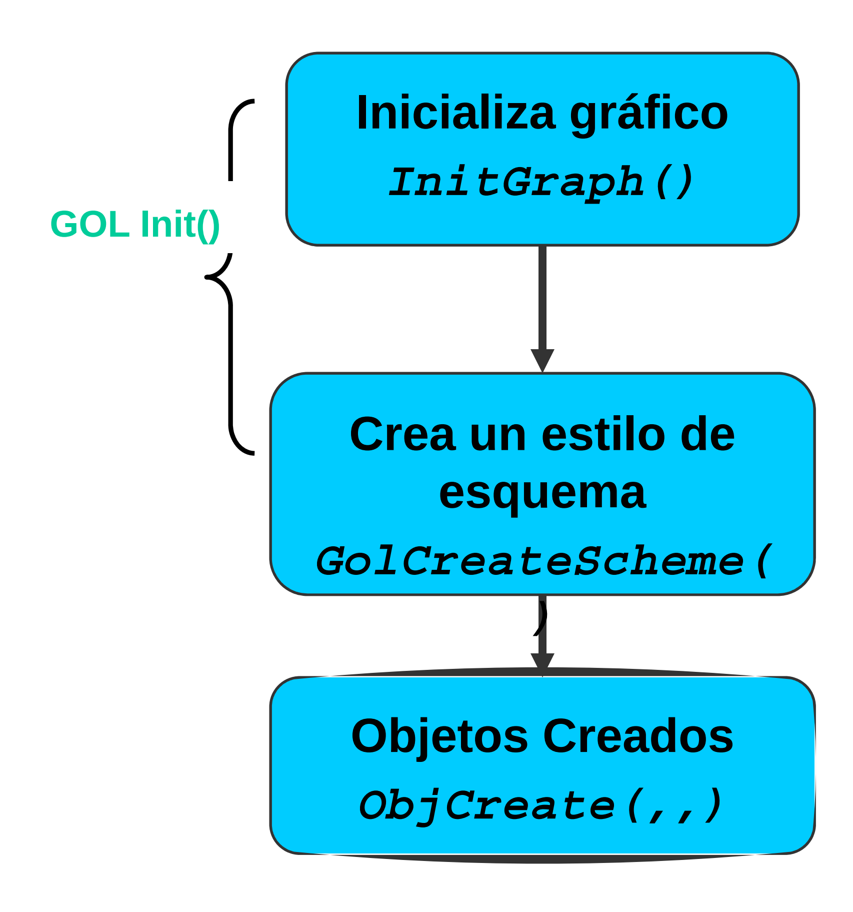
- Por defecto se usa el esquema si no esta definido uno propio

```
typedef struct {
   WORD EmbossDkColor;
   WORD EmbossLtColor;
   WORD TextColor0;
   WORD TextColor1;
   WORD TextColorDisabled;
   WORD Color0;
   WORD Color1;
   WORD ColorDisabled;
   WORD CommonBkColor;
   char *pFont;
} GOL_SCHEME;
```





Uso de la Librería Gráfica

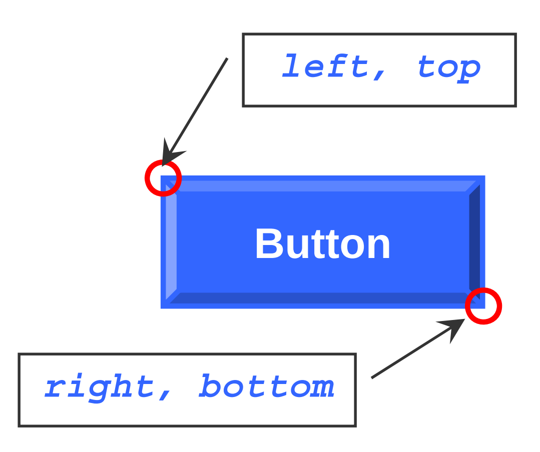




Ejemplo de un Boton

- API definida para BtnCreate

```
BUTTON *BtnCreate(
    WORD   ID,
    SHORT
           left,top,
           right,bottom,
          state,
    char *pBitmap,
          *pText,
    GOL_SCHEME *pScheme
)
```





Ejemplo de un Boton

- **Boton** `state` **pueden ser cualquiera de los siguientes**

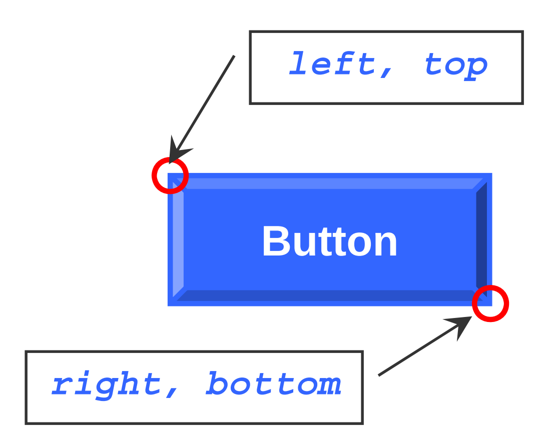
```
#define BTN_FOCUSED   0x0001  // bit for focus state (dotted
                              // rectangle will be drawn)
#define BTN_PRESSED   0x0002  // bit for press state (button
                              // is pressed)
#define BTN_DISABLED  0x0010  // bit for enable state
                              // (disabled button has separate
                              // colors)
#define BTN_DRAW      0x4000  // bit to indicate button should
                              // be redrawn
#define BTN_REMOVE    0x8000  // bit to indicate button should
                              // be removed from screen
```



Ejemplo de un Boton

- **Para crear un boton, se necesitan los siguientes perámetros:**

```
BUTTON *BtnCreate(
   WORD   ID,
   SHORT
          left,top,
          right,bottom,
         state,
   char *pBitmap,
         *pText,
   GOL_SCHEME *pScheme
)
```



*pBitmap – **Setear para NULL cuando no se usan bitmaps.**

*pScheme – **Define el uso o setea para NULL para usar seteos por default.**



Graphics Library Usage

GOL Init()

Initcializa el gráfico
`InitGraph()`

Crea el estilo del Objeto
`GolCreateScheme()`

Crea Objetos
`ObjCreate(,,)`

Dibuja objetos
`GOLDraw()`



Ejemplo de un Boton

- **Para presentar el boton sobre la pantalla llamar a `BtnDraw(pB)`.**
  - **pB puntos del Boton a ser presentado.**
- **Para manejar el estado del Boton**

```
SetState(pB, stateBits)
ClrState(pB, stateBits)
GetState(pB, stateBits)
```

`stateBits` son bits de estado variable



Ejemplo de un Boton

Otros soportes APIs

- **Maneja texto:**
  - *BtnGetText(pB)* & *BtnSetText(pB,*)*
- **Maneja Bit Map:**
  - BtnGetBitmap(pB) & *BtnSetBitmap(pB, *)*
- **Maneja el estilo del esquema:**
  - *GetScheme()* & *SetScheme()*



Uso de la Librería Gráfica

GOL Init()

Inicializa gráficos
InitGraph()

Crea el estilo del esquema
GolCreateScheme()

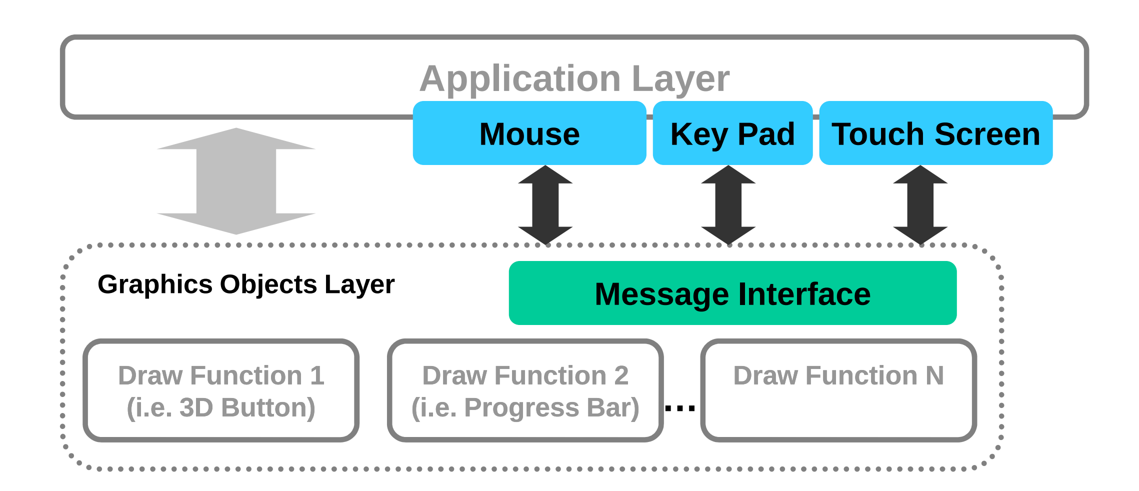
Crea el objeto
ObjCreate(,,)

Dibuja el objeto
GOLDraw()

Get User Inputs
Message Struct



Message Interface



- **Para una mejor experiencia del usuario, el dispositivo de entrada del usuario debe trabajar unido a la exhibición en el display**
- **Simplifica la forma de manejo del objeto desde el punto de vista de la aplicación**
  - **La cara de la interfaz de mensajes simplifica la integración de los dispositivos de entrada del usuario**
  - **Soporte para una variedad de dispositivos de entrada**
  - **Integración inconsulta con los objetos gráficos**



Mensajería

- Estructura de un mensaje

```
typedef struct {
BYTE        type;
BYTE                event;
SHORT       param1;
SHORT       param2;
} GOL_MSG;
```

- *type* define un tipo de entrada la cual determina como *param(n)* son interpretados
  - Ejemplo: Mensaje desde el touch screen:
    - *param1* : x position
    - *param2* : y position
  - Para keypad: los parámetros pueden ser un código de una tecla



Mensajería

- `event` es un mensaje genérico de un dispositivo de entrada
  - `event` definido para un touch screen
    - EVENT_PRESS
    - EVENT_RELEASE
    - EVENT_MOVE
  - `event` definido para un teclado puede ser
    - EVENT_PRESS
    - EVENT_RELEASE



Mensajería

- **Acción de usuario ID**
  - **Lista de acciones definidas (o cambio de estado) que un objeto ha aceptado.**
  - **Cada tipo de objeto tiene una lista predefinida.**
    - **Ejemplo de un boton:**
      - *BTN_MSG_PRESSED*
      - *BTN_MSG_RELEASED*
  - **Si el objeto no es afectado por la acción del usuario este retorna *MSG_INVALID*.**



Uso de la Librería Gráfica

GOL Init()

Inicialización de gráficos
InitGraph()

Estilo de esquema Creado
GolCreateScheme()

Crea Objetos
ObjCreate(,,)

Dibuja Objetos
GOLDraw()

Get User Inputs
Message Struct

Procesador de Mensajes de Usuario
GOLMsg()



Uso de la librería Gráfica

GOL Init()

Inicializa Gráficos
InitGraph()

Crea un estilo del esquema
GolCreateScheme()

Crea Objetos
ObjCreate(,,)

Dibuja Objetos
GOLDraw()

Get User Inputs
Message Struct

Procesador de Mensajes de Usuario
GOLMsg()



Fundamentos del Touch Screen



Touch Screen Resistivo

Corte Transversal

Hard Coat

Seal

Space

Anti Glare/(Circular Porlalizer)

Glass or Polyester

ITO

Glass



Touch Screen Resistivo

- El mas popular en uso industrial y aplicaciones consumo
- Es el mas usado por su relación costo y facilidad de uso



Detección de un Touchscreen de 4 cables

PIC® MCU w/ADC

Digital I/O

Digital I/O

Dig-O/AN-I

Dig-O/AN-I



Detección de un Touch Screen de 4 cables



Detección de un Touchscreen de 4 cables

Los Pines del ADC pueden actuar como una entrada analógica o como una salida digital



4-wire Touch Screen Detection

PIC® MCU w/ADC

Digital I/O

Digital I/O

Dig-O/AN-I

Dig-O/AN-I

0

1

Sense

Vdd



Detección del Touch (4-cables)

PIC® MCU w/ADC
Digital I/O
Digital I/O
Dig-O/AN-I
Dig-O/AN-I
0
1
Sense
Vdd



Conección Universal 4/5-Cables



Conexión de Filtros 4- y 5-Cables

X+/LR
47
X-/UR
NC/LL
Y+/UL
Y-/Wiper
Dig-O
Dig-O/ AN-I
Dig-O
Dig-O
Dig-O/ AN-I
Universal 4/5 wire connector
10 nF

Do = Digital output
Ai = Analog input



Calibración



Calibración

100
0
0
100
P

$X_p=50$
$Y_p=50$

$X_p=75$
$Y_p=65$



Calibración

P2 $_{x2,y2}$

P3 $_{x3,y3}$

P1 $_{x1,y1}$

=AX+BY+C

=DX+EY+F

A, B, C, D, E, y F son constantes.

•X, Y son ccordenadas medidas.

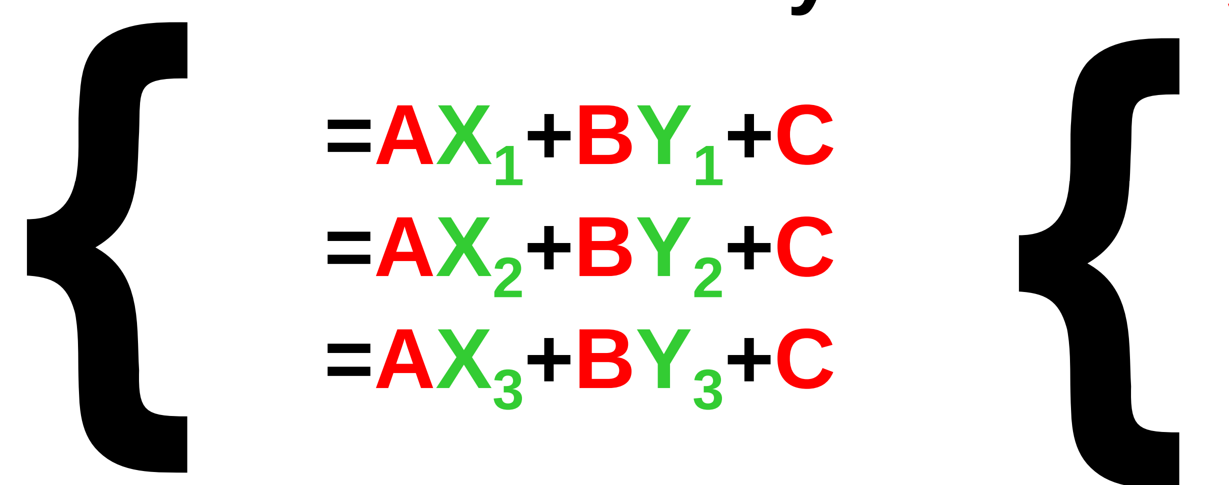
Microchip
Regional Training Centers

Calibración

Measured/ coordinates

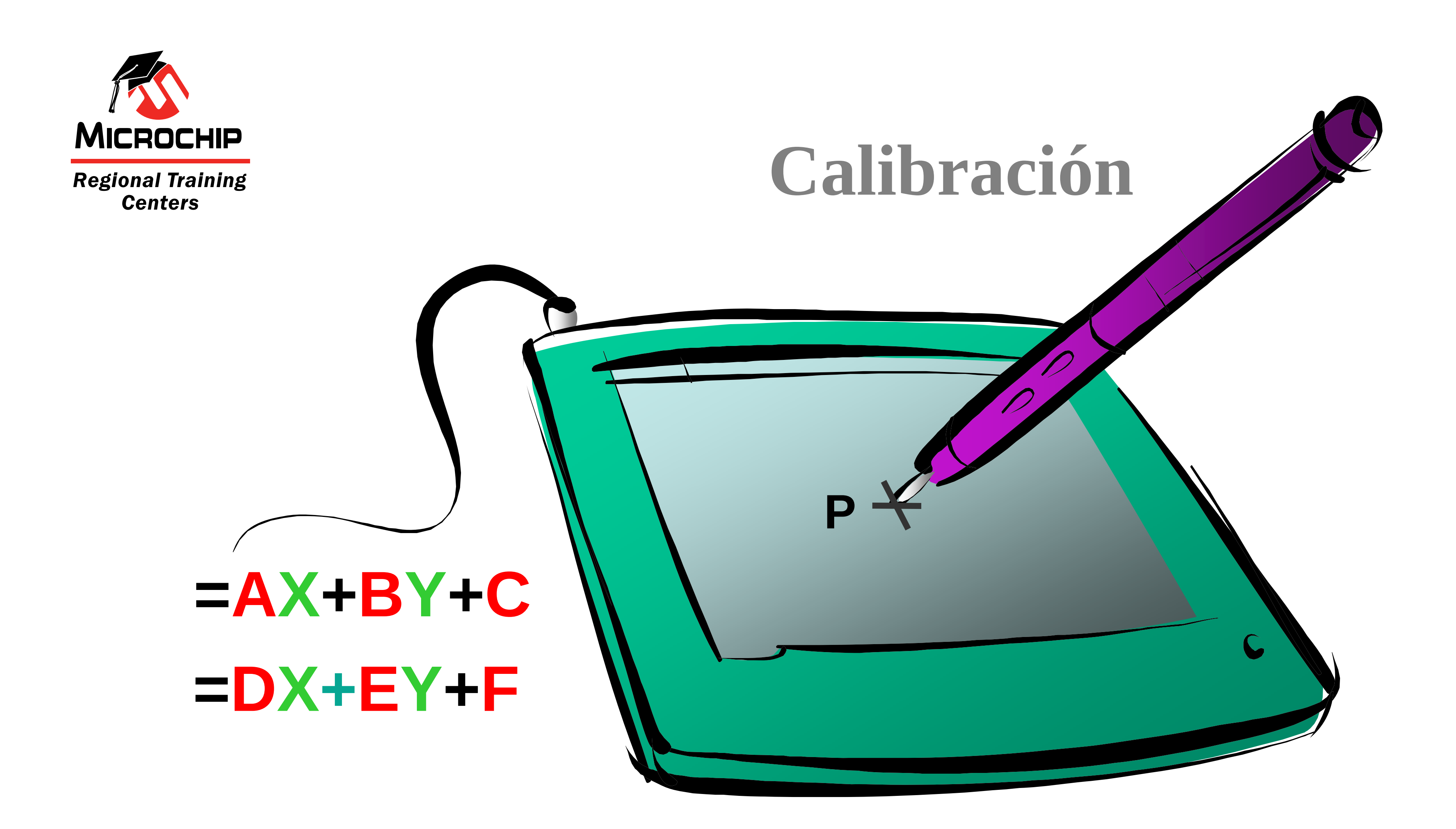
P2 x2,y2
P1 x1,y1
P3 x3,y3

- Seis ecuaciones y seis incógnitas



$$=AX_1+BY_1+C$$
$$=AX_2+BY_2+C$$
$$=AX_3+BY_3+C$$

$$=DX_1+EY_1+F$$
$$=DX_2+EY_2+F$$
$$=DX_3+EY_3+F$$



- X, Y son coordenadas medidas
- A, B, C, D, E, y F son las constantes que calculábamos



Análisis de la versión parcial de un programa



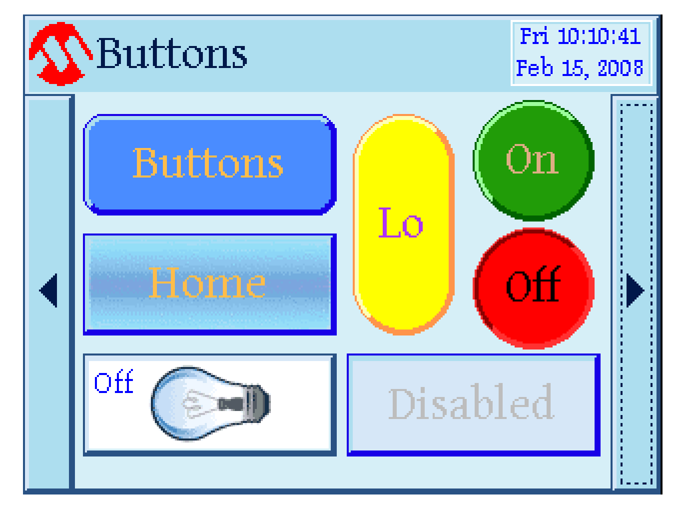
Basic Graphics Object Layer Demo

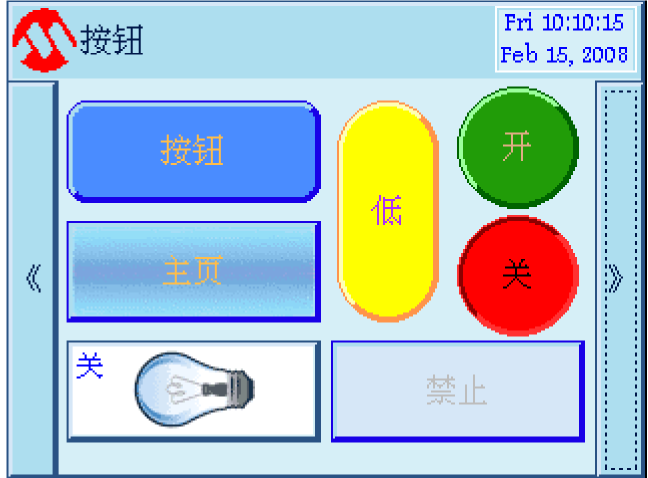
- Codigo fuente provisto con la librería
- Demostraciones de aplicaciones multi-pantalla
- Acepta variedad de entradas de usuario
  - Variar la forma de onda con potenciómetro
  - Switch S4 para seleccionar items, S5 como "enter"
  - Touchscreen via yema del dedo o de una punta
- Demostraciones de animaciones compatibles
- Usa 8-bit PMP sobre PIC MCUs



Ejemplo de memoria requerida para gráficos

- Espacio en Flash (10.6 KBytes)
  - Stdlib ~8 KBytes
  - Boton ~1 KBytes
  - Ventana ~0.8 KBytes
  - Texto estático ~0.7 KBytes
- Espacio en SRAM (130 Bytes)
  - Stdlib ~110 Bytes
  - Boton 8 Bytes
  - Ventana 8 Bytes
  - Texto estático 2 Bytes
- Espacio Dinamico (287 Bytes)
  - 9 Botones 243 Bytes
  - 1 ventana 23 Bytes
  - 1Texto estático 21 Bytes







Graficos plus demoUSB

- Código fuente disponible sobre pedido
- Integra USB Host stack con Librería Gráfica
  - RGB Lighting board shows HID
  - USB Stick shows Mass Storage Device
    - FAT16/32 file system implemented
- Usa controlador externo SSD1906 LCD
  - EEPROM conectada al PMP almacena iconos



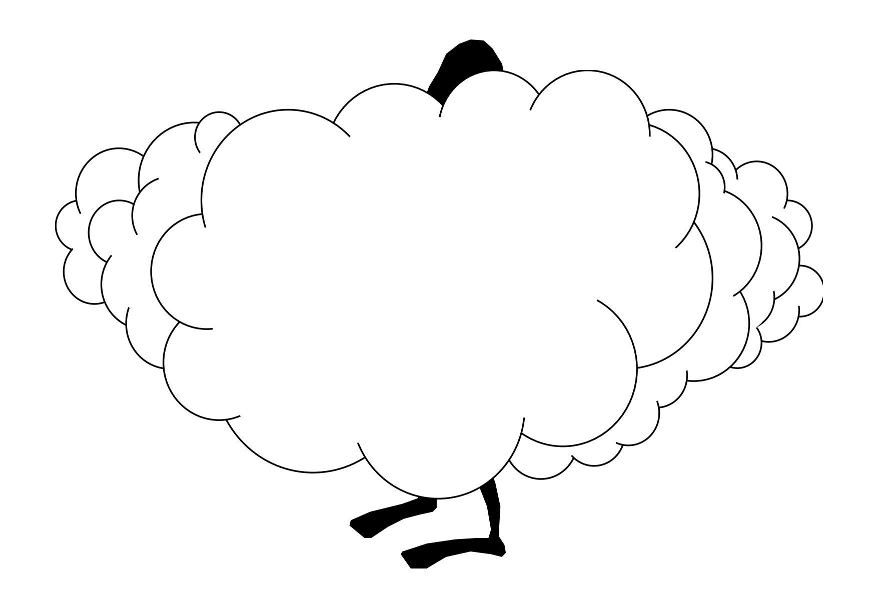
Regional Training Centers

Magic Touch



Filosofía de la Competencia

- Patentes sobre hardware de medida de capacitores
- Software es “stock” bloque comun del codigo para el touch
- Conjeturan que el sistema es mejor que lo que necesitan los clientes





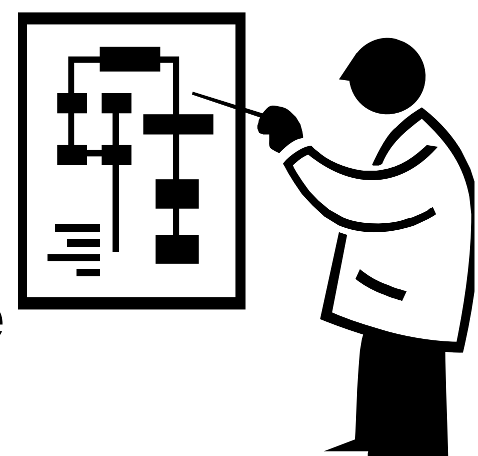
Filosofía del MTouch

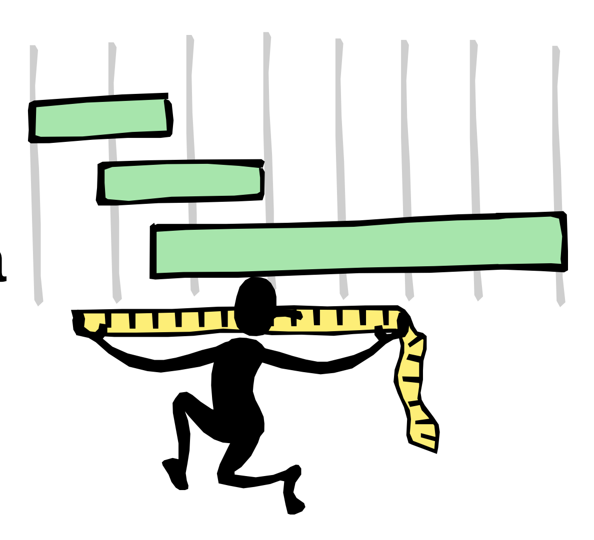
- Enseñar a los clientes a desarrollar sus propias soluciones
- Software Developers Kit (SDK)
- Herramientas para desarrollo y diagnóstico
- clientes
  "consiga lo que usted quiere", no "tome lo que puede conseguir."



Herramientas

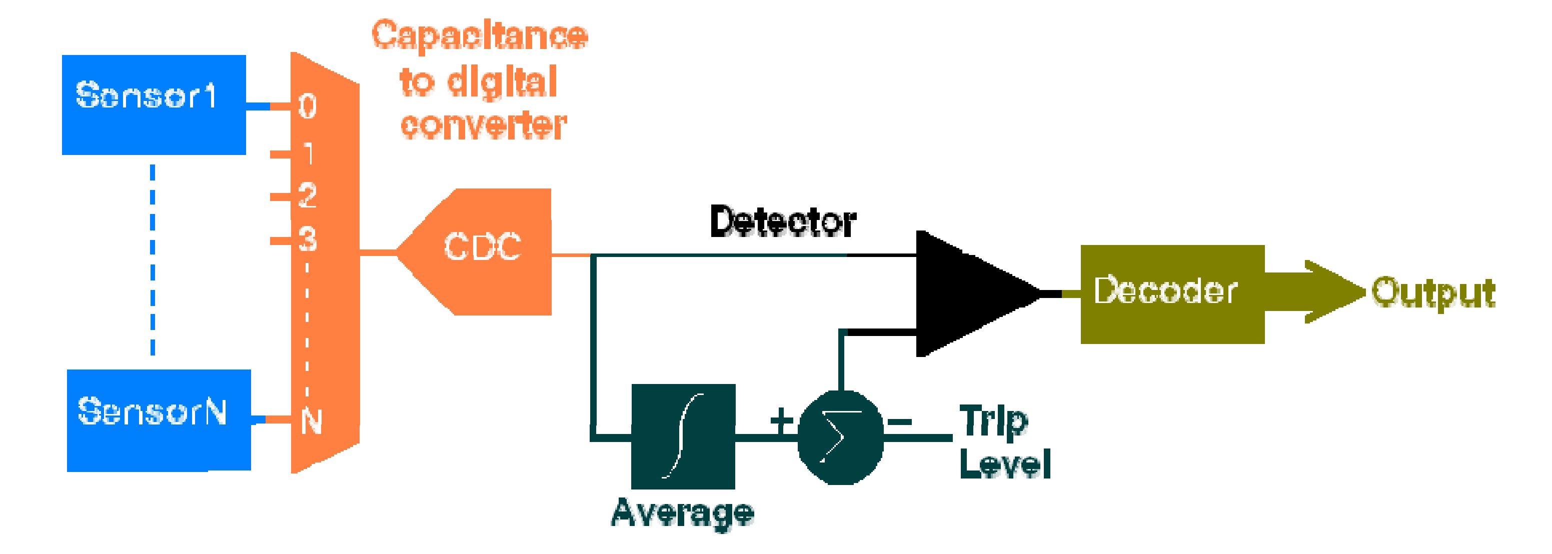
- Software Developer's Kit
  - Bloques vacios para un sistema touch
  - Manual de Usuario con ejemplos de diseño
- Touch GUI
  - Evalua un nuevo diseño touch
  - Determina la resolución requerida
  - Determina la sensibilidad requerida
  - Evalua factores del ambiente







Software Kit de desarrollo

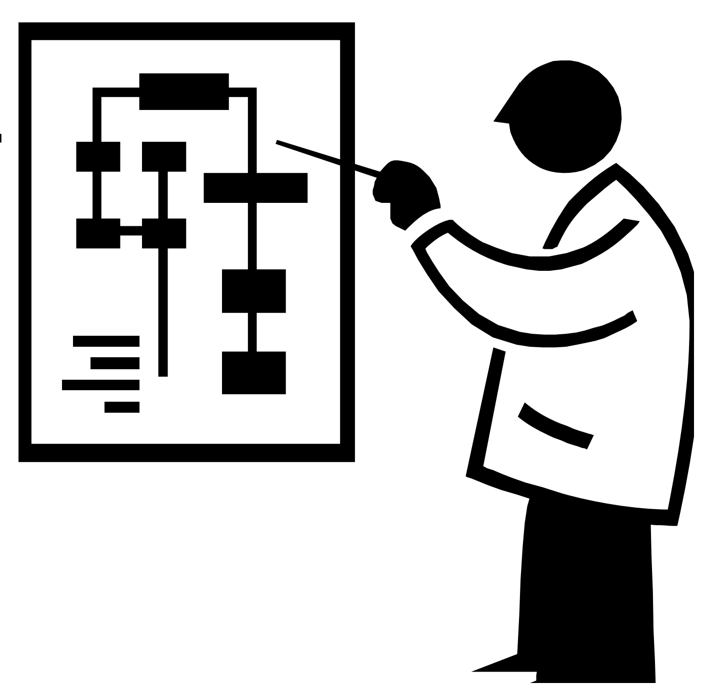


- **El sistema Touch es segregado en bloques**
- **Los bloques optimizados para aplicaciones específicas se proporcionan**
- **Bloques organizados en Librerias de Módulos de Código**



SDK Manual de Usuario

- **Users Manual**
  - **Proporciona “el como” crear una interfaz touch**
  - **Documentación del bloque de código**
  - **Hardware requerido para cada bloque**
  - **Combinación de bloques recomendada**
  - **Referencias de diseño para ambientes específicos.**

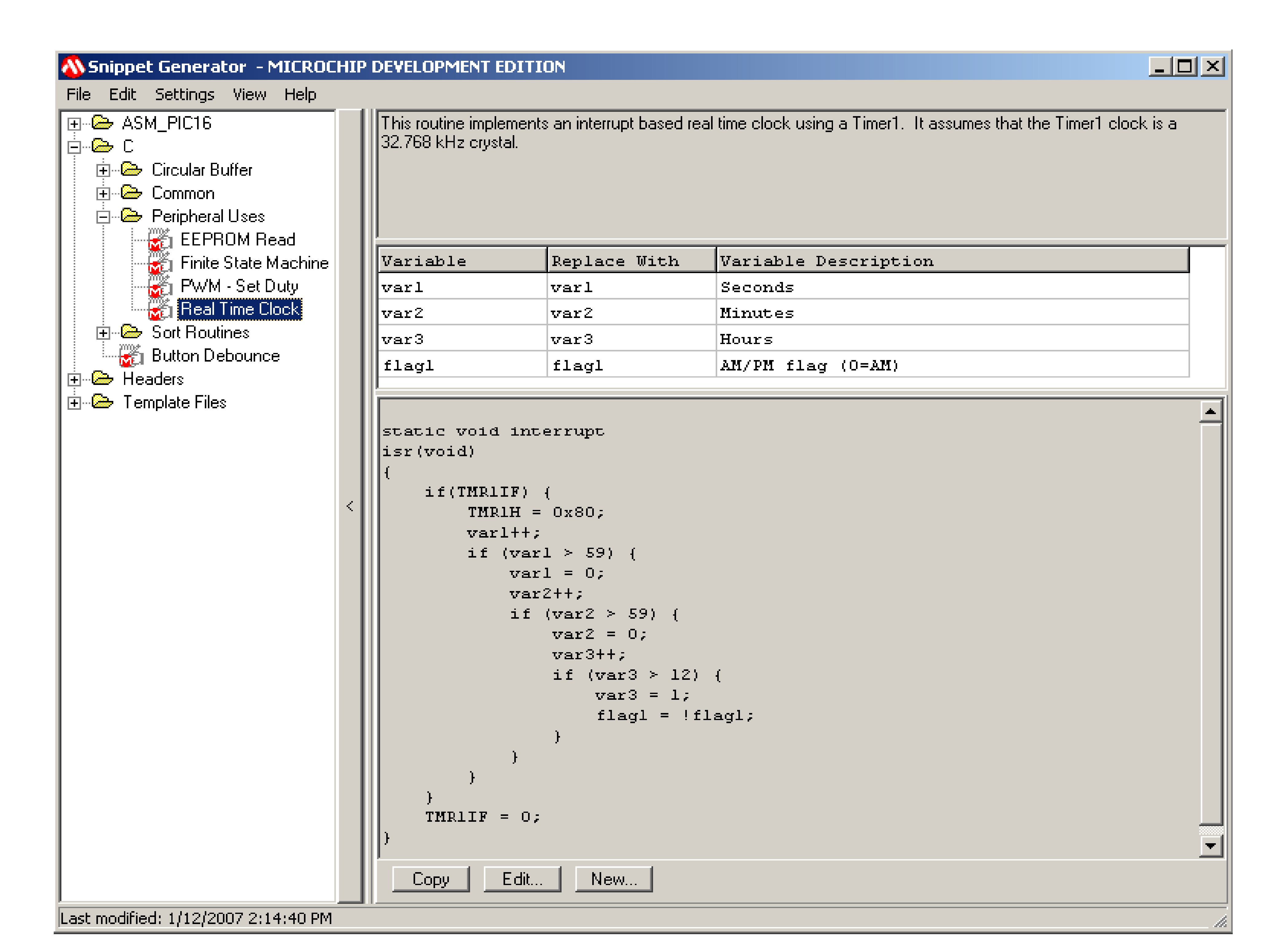




Librería del Bloque de Código

- Organización del Bloque de Código y customización de herramientas
- Organización estilo carpeta
- Función de substitución de Nombre y Carpeta
- Trabajando con un editor de texto
- Lenguaje independiente

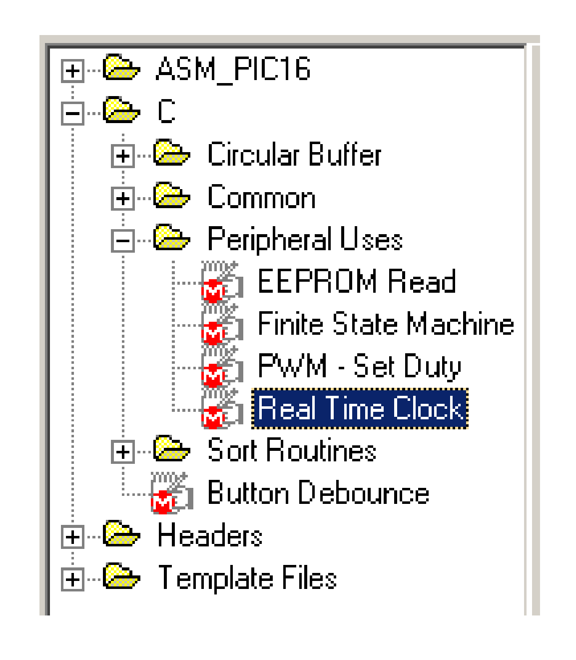



Snippet Generator - MICROCHIP DEVELOPMENT EDITION
File Edit Settings View Help
ASM_PIC16
C
Circular Buffer
Common
Peripheral Uses
EEPROM Read
Finite State Machine
PWM - Set Duty
Real Time Clock
Sort Routines
Button Debounce
Headers
Template Files
This routine implements an interrupt based real time clock using a Timer1. It assumes that the Timer1 clock is a 32.768 kHz crystal.
Variable | Replace With | Variable Description
var1 | var1 | Seconds
var2 | var2 | Minutes
var3 | var3 | Hours
flag1 | flag1 | AM/PM flag (0=AM)
static void interrupt
isr(void)
{
if(TMR1IF) {
TMR1H = 0x80;
var1++;
if (var1 > 59) {
var1 = 0;
var2++;
if (var2 > 59) {
var2 = 0;
var3++;
if (var3 > 12) {
var3 = 1;
flag1 = !flag1;
}
}
}
}
TMR1IF = 0;
}
Copy
Edit...
New...
Last modified: 1/12/2007 2:14:40 PM

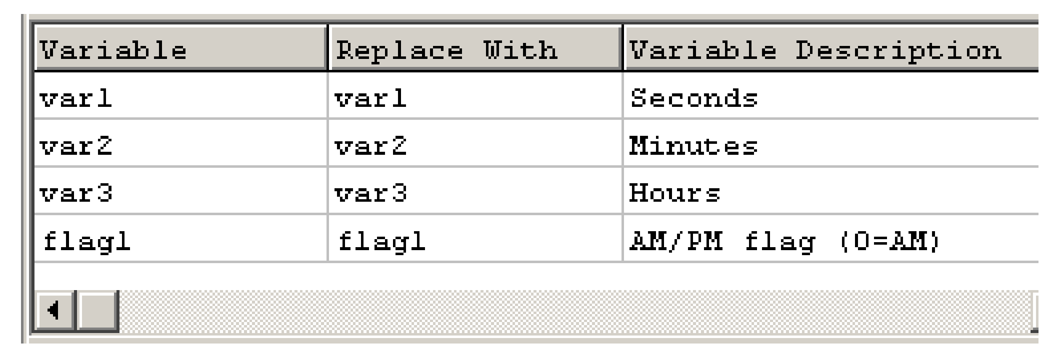


Usando un bloque de código

- Selección del Boque de Código



- Renombre de variables

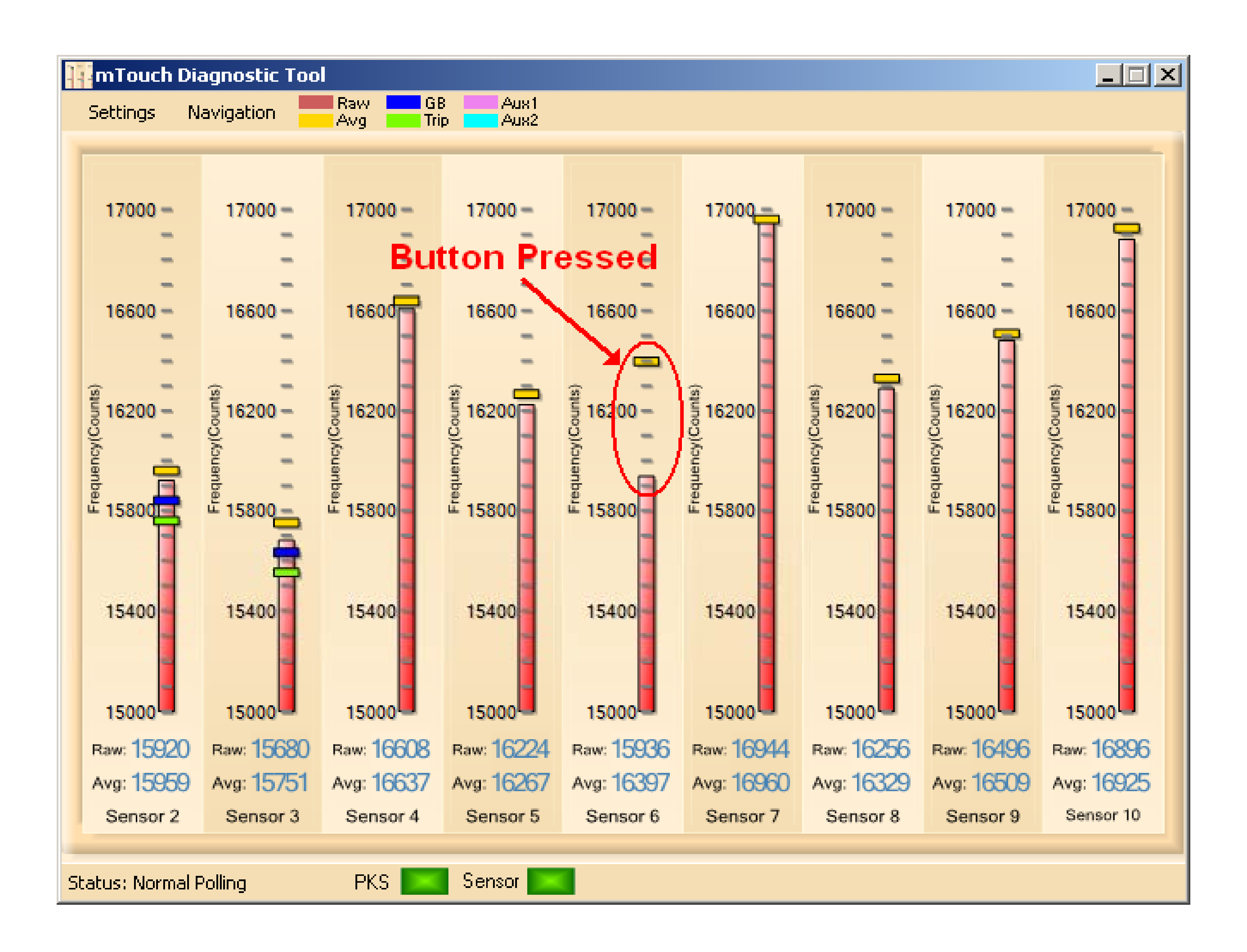


Variable	Replace With	Variable Description
var1	var1	Seconds
var2	var2	Minutes
var3	var3	Hours
flag1	flag1	AM/PM flag (0=AM)



- Copiar y pegar



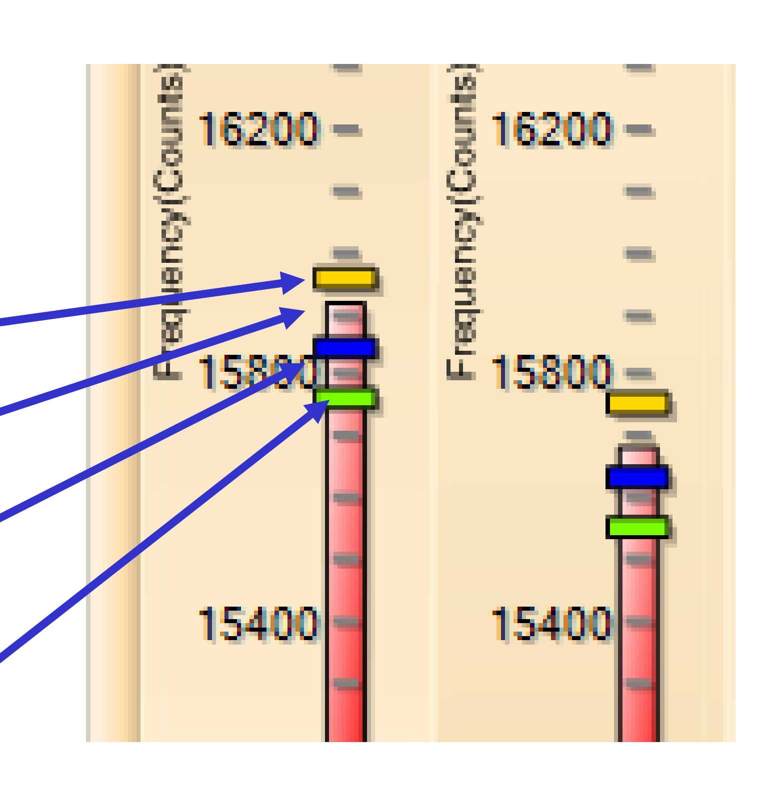


mTouch Diagnostic Tool
Settings
Navigation
Raw
Avg
GB
Trip
Aux1
Aux2
Button Pressed
Frequency(Counts)
17000
16600
16200
15800
15400
15000
Raw: 15920
Avg: 15959
Sensor 2
Raw: 15680
Avg: 15751
Sensor 3
Raw: 16608
Avg: 16637
Sensor 4
Raw: 16224
Avg: 16267
Sensor 5
Raw: 15936
Avg: 16397
Sensor 6
Raw: 16944
Avg: 16960
Sensor 7
Raw: 16256
Avg: 16329
Sensor 8
Raw: 16496
Avg: 16509
Sensor 9
Raw: 16896
Avg: 16925
Sensor 10
Status: Normal Polling
PKS
Sensor



MTouch GUI

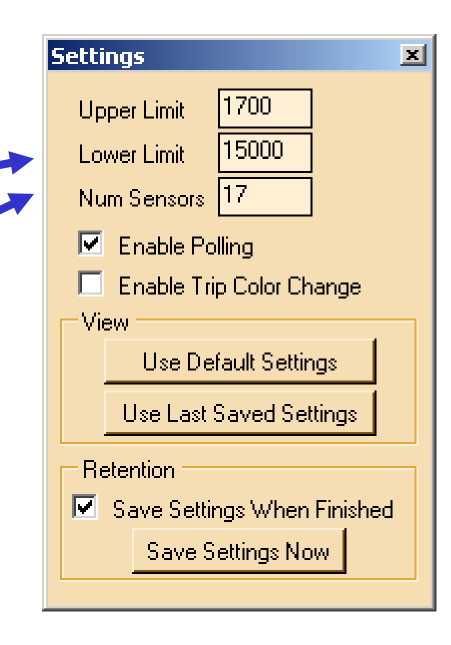
- Cada Boton y circuito es diferente
- Promedio del sensor
- Valor actual crudo
- Banda protectora para el promedio bloqueado gated
- El Nivel del recorrido es el umbral



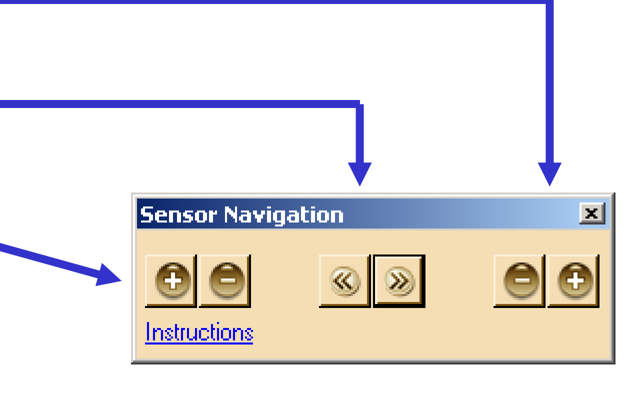


MTouch GUI

- Settings de Ventana
  - Muestra Límites
  - Número de sensor



- Navegación del Sensor
  - Add/remove desde la derecha
  - Mover izquierda or derecha
  - Add/remove desde la izquierda





Demo 1

- Observe el comportamiento de las placas preprogramadas
- Conecte el PICkit Serial Analyzer (PKSA)
- Arranque el MTouch GUI usando PICkit Serial

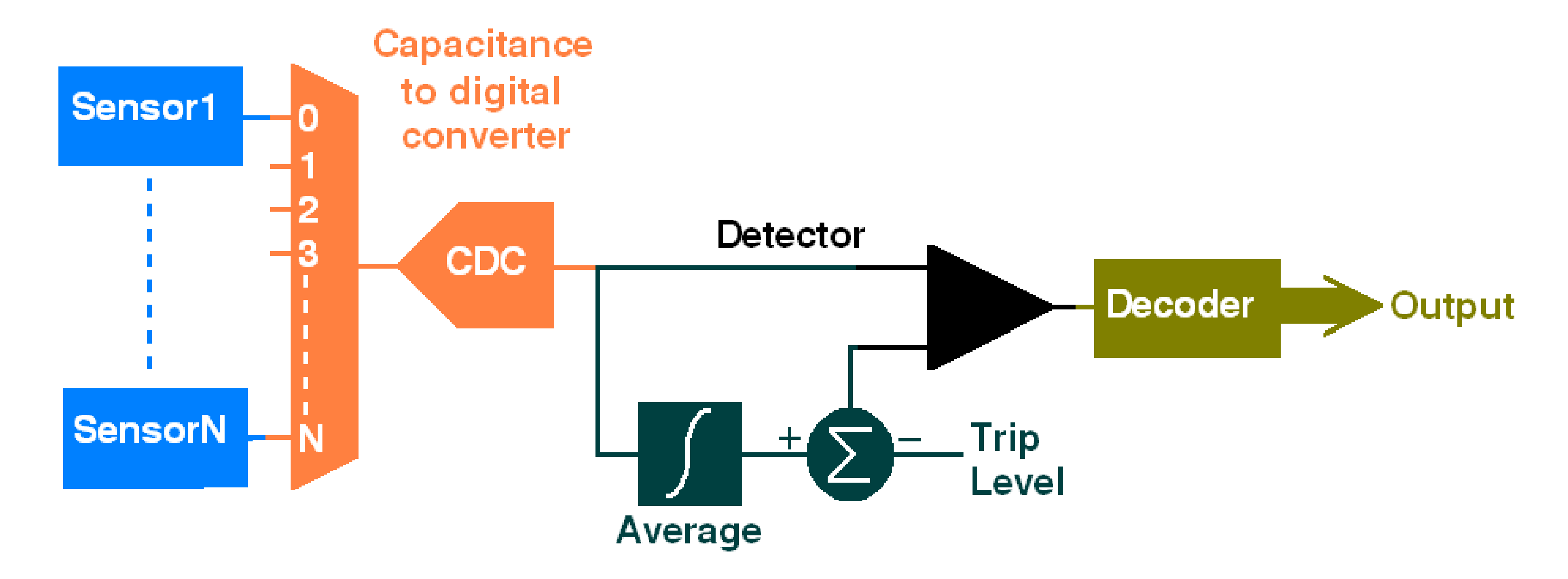


Revisión del Demo

- Son los valores medios iguales para todos los botones?
- Hasta donde el valor crudo cambio durante un touch suave?
- Hasta donde el valor crudo cambio durante un touch pesado?
- Había una interacción entre los botones?
- Que recorrido de niveles usted sugeriría?



Diagrama en Bloques de un Sistema



- Un sistema touch es mostrado arriba
- Sensores y CDC son sobre todo hardware
- Detectores y Decodificadores son software



Hardware de Conversión

- Oscilador de frecuencia variable basado en un comparador
- Contador de frecuencia basado en un timer



Multiplexación de Botones

Analog Multiplexer

Mux1 0
Mux2 1
Mux3 2
Mux4 3
Mux5 4
Mux6 5
Mux7 6
Mux8 7

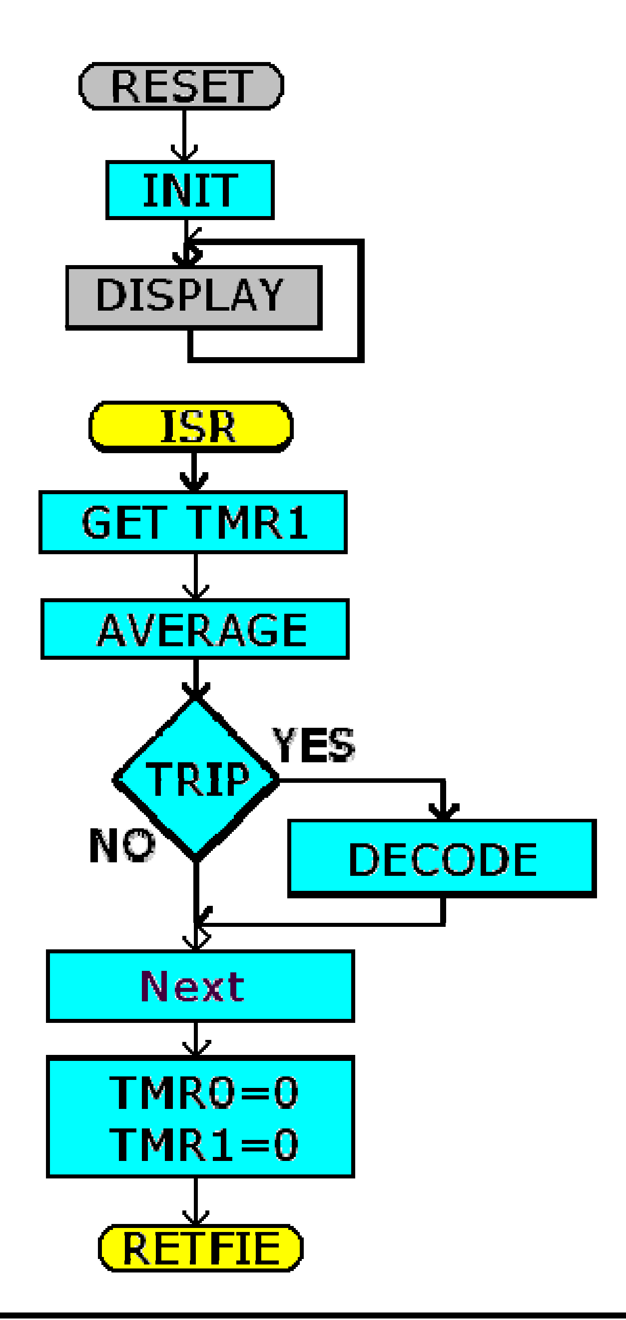
Sel0
Sel1
Sel2
Out

+5V
R1
R2
Cvref
C1
C2
S Q
R Q̄
TIMER1
TIMER0
R4
R5
R6
R7
Ctouch1
Ctouch2
Ctouch3



Software

- Main Loop
  - Inicializa periféricos
- TMR0 ISR
  - Consigna el valor del TMR1
  - Promedio
  - Compare el recorrido
  - Decodificar
  - Set nuevo boton
  - Reiniciar TMRs





Consiga el valor

- Si borra la interrupción...
  - Poniendo en off TMR1
  - Consiga el Valor
  - Concatenando valores
  - Almacenando en array CRUDO
- Si INT durante una Interrupción...
  - Borra los timers
  - RETFIE



Promediando

- El estilo del Filtro RC requiere un promedio justo de dos bytes.
  - + DIFF = button press
  - - DIFF = button release

```
DIFF = AVG – (LAST * 16)
DIFF = DIFF / 16
AVG  = AVG + DIFF
```

- Hacer un promedio lento
  - Haga un promedio de cada 8vo valor
- Haciendo un promedio bloqueado
  - Promediar si delta < a la banda de Guarda



Encontrar el valor siguiente

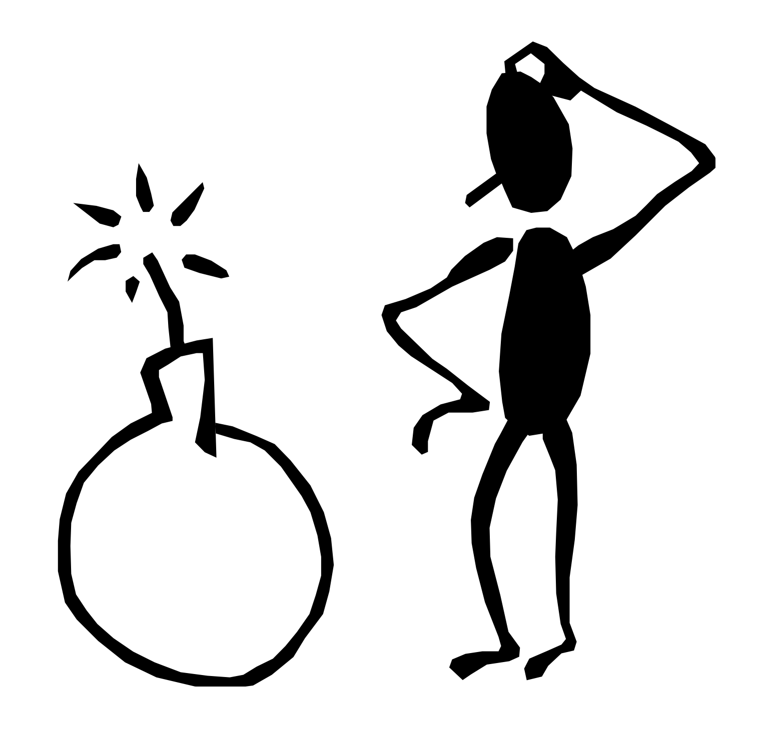
- Basado sobre una indexación de valores de datos internos
  - Update el multiplexor externo y las salidas
  - Update el multiplexor interno y pone el SFR





Decodificando

- Decodificación basica
  - El promedio bloqueado da una indicación on-off del touch
  - El promedio lento requiere un set-reset estilo de rebote
- Tiempo promedio del boton pegado
- Tiempo promedio disparado
- Pressed at powerup average resets





Implementaciones con Microcontroladores de bajo costos

- PIC16F61X son alternativas de muy bajo costo
- Sin embargo;
  - No tienen periférico SSP
  - el ICD afecta al sensor capacitivo
- Tan;
  - GUI is no esta disponible para testear botones
  - Los niveles de recorrido son mejores que los esperados



Touch Proporcional

- Recorrido de las Bases sobre el percentaje de desplazamiento en frecuencia
- Quita la mayor parte de la variación del sensor capacitivo
- El Algorítmo ha sido optimizado para reducir la matemática
- Cuando se combina con lo elegido, puede compensar parcialmente la humedad sobre los botones
- Solo se puede presionar un boton por vez



Lectura del valor

- El porcentaje de desplazamiento es 0-20%
- Para evitar el uso de punto flotante, los valores son multiplicados por 1000
  - 14.2% (0.142) x 1000 produce 142
- Los desplazamientos positivos y negativos pueden ser considerados separadamente



Promediando

- El estilo del filtro RC requiereun promedio justo de dos bytes.
  - + DIFF = button press
  - - DIFF = button release

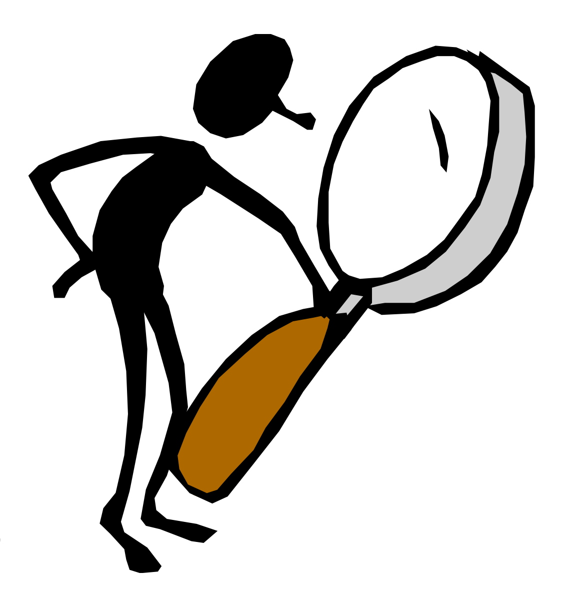
```
DIFF = AVG – (LAST * 16)
DIFF = DIFF / 16
AVG  = AVG + DIFF
```

- Promediando lentamente
  - Promedio cada 8vo valor
- Promediando Gated
  - Promedio si delta < Banda de Guarda



Votación

- Util para el agua en los botones
- Los porcentajes de presionado son clasificados de mayor a menor
  - La mayor presión es el boton“mas presionado”
- On Rotary and Slider, top two are overlapped buttons
  - Ratio of over lap is position between buttons





Touch Durante Sleep

- **En el modo sleep, no hay Fosc, tampoco T0IF**
- **Hay dos opciones**
  - **1. Use WDT en lugar de TMR0**
    - **Requiere calibración individual del oscilador del WDT (Impráctico)**
  - **2. Despertar periódicamente y medir un boton al despertar**
    - **Require Botones simples durante el modo sleep**
    - **La sensibilidad del boton puede ser incrementada para atender la ondulación de la mano**



Option 2

- La siguiente función es modificada con un Modo Sleep flag pra la operación de un boton simple

- Sleep y Wakeup para decodificar el comando es agregado un loop en el código principal

- If applicable, alternate TRIP value would be required for HandWave wakup



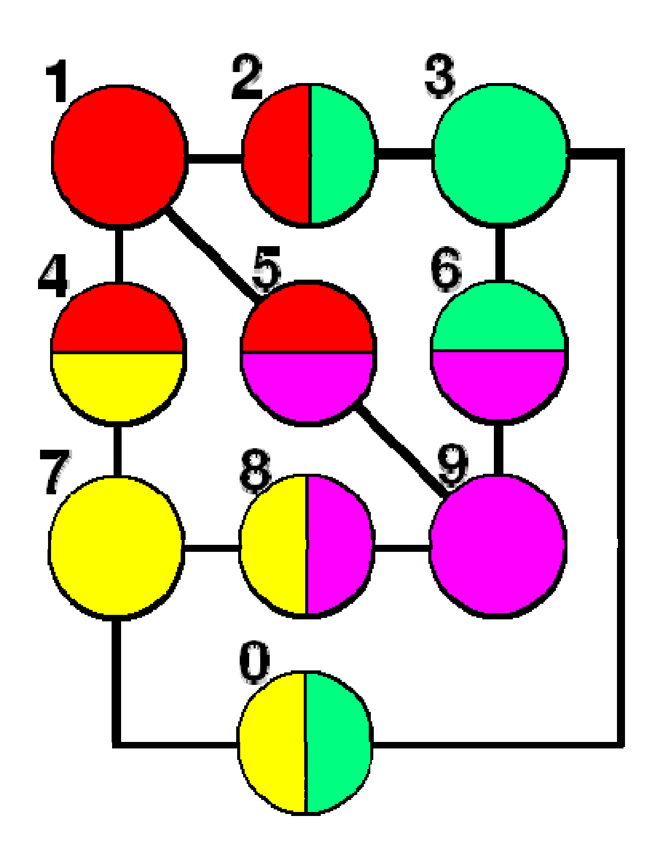
Presionado Apareado

- El presionado apareado combinado en botones simples para crear nuevos botones
  - El desplazamiento en frecuencia del presionado apareado *puede ser muy lento* requiriendo valores secundarios de recorrido
  - El presionado apareado no es compatible con el comando del boton Chord
- Para decodificar, chequear para saber si hay apareados, entonces simple



Metodo del "Presionado Apareado"

- Amplia de 4 a 10 botones
- 1, 3, 7, & 9 son los botones enteros
- 2, 4, 5, 6, 8, & 0 son los botones apareados
  - El presionado apareado solo produce la ½ de capacidad
  - Requiere scan de todos los botones para una decodificación valida
  - No se puede diferenciar el presionado de dos botones de un presionado apareado





Summary



Resumen

- Arquitectura de la librería gráfica de Microchip
- Como usar los objetos gráficos (aparatos)
- Fundamentos del touch screen resistivo e integración en aplicaciones



Resumen

- Soluciones para MCU PIC® gratis
- Seamless integration con dispositivos de entrada de usuario
- Alta optimización (Velocidad y Tamaño)
  - Stdlib + Primitive + Device Driver – Approx 8KB PM, 52 bytes of RAM
  - Cada objeto gráfico – Approx 1KB PM, 10 – 15 bytes RAM
  - Cada Font – Approx 5 – 7 KB of PM
  - Memoria dinámica – Varia basado en el contenido de los gráficos. Refiera a las notas de lanzamiento



Conclusiones

- La mayor parte del touchcapacitivo es software
- MTouch incluye una GUI y SDK
- El GUI ayuda en la evaluación y diseño de la interfaz
- El SDK crea la interfaz



Trademarks

Traducido por: Andrés Raúl Bruno Saravia
Prof. de Electrónica
Instituto Técnico OHM